

Applicant : William Lin
Serial No. : 10/665,156
Filed : September 18, 2003
Page : 11 of 16

Attorney's Docket No.: 07844-610001 / P563

BEST AVAILABLE COPY

Amendments to the Drawings:

The attached replacement sheet of drawings includes changes to Fig. 1 and Fig. 2 and replaces the original sheet including Fig. 1 and Fig. 2.

In Figure 1, "Prior Art" has been added.

In Figure 2, "Prior Art" has been added.

Attachments following last page of this Amendment:

Replacement Sheet (1 page)
Annotated Sheet Showing Change(s) (1 page)

REMARKS

Claims 1-30 are currently pending, of which claims 1, 13, 16, and 28 are independent. Claims 7, 12-13, and 27-28 are amended to correct typographical errors. Reconsideration of the action mailed on March 29, 2006, is respectfully requested in light of the foregoing amendments and the following remarks.

The examiner rejected claims 1-30 under 35 U.S.C. § 102(e) as allegedly anticipated by U.S. Patent Application Publication 2004/0177315 ("Brown").

Information Disclosure Statement

The examiner stated that the information disclosure statement filed October 25, 2004, failed to comply with 37 CFR 1.98(a)(2). Specifically, the examiner stated that the submitted reference was illegible. A legible copy of the cited reference is being submitted with this response. The applicant respectfully requests that the Examiner consider, initial, and return the form PTO-1449 that was filed on October 21, 2004.

Objections to the Drawings

The examiner objected to FIGS. 1 and 2 under MPEP § 608.02(g) as failing to be designated as "prior art". The applicant has enclosed corrected drawings designating FIGS. 1 and 2 as prior art. The applicant respectfully submits that the examiner's objection has been overcome.

Objections to the Specification

The examiner objected to the specification under MPEP § 608.01 as including embedded hyperlink references. The applicant has amended the specification to remove the hyperlinks. The applicant respectfully submits that the examiner's objection has been overcome.

Section 102 Rejections

Claim 1 stands rejected over Brown. Claim 1 recites a computer-implemented method for correcting a structured electronic document. A structural aspect of the structured electronic document is identified as failing to conform to rules of a markup language format. The rules are

associated with the structured electronic document. One or more changes are suggested to the user that would correct the identified structural aspect. User input is received selecting a change of the suggested changes. The selected change is applied to the structured electronic document.

Brown does not disclose or suggest identifying of a structural aspect as failing to conform to rules of a markup language format. Brown discloses a technique for restricting changes that can be made to contents of a structured document. *See Abstract.* A bounding file is used to define restrictions on how a particular structured document can be manipulated. *See paragraph 23.* The bounding file specifies one or more elements of the structured document that can be edited as well as elements that should be hidden from editing. *See paragraph 25.* Editing by the user is therefore constrained by the bounding file. *See paragraph 26.*

The examiner states that Brown discloses the claimed identifying of a structural aspect as failing to conform to rules of a markup language format at paragraph 50. The applicant respectfully disagrees. Paragraph 50 of Brown reads, in pertinent part, as follows:

For example, a bounding file might specify that certain tags (including their data and the data of their child tags) are never editable, while another bounding file might specify that tags are non-editable only under certain conditions (such as the tag's attributes having particular values). The bounding files are also encoded in XML, in preferred embodiments, and are referred to equivalently herein as "XML bounding files".

The cited paragraph states that the bounding file specifies tags that cannot be edited or tags that can be edited under particular conditions. However, the cited paragraph does not disclose or suggest any failure of a structural aspect to conform to rules of a markup language. Additionally, there is no disclosure or suggestion in the cited paragraph of identifying any such structural aspects as failing to conform to any rules. Furthermore, Brown is concerned with limiting user editing of a document, which could generate new errors, not the identification of existing errors.

Additionally the examiner states that Brown discloses suggesting one or more changes to a user to correct the identified structural aspect that fails to conform at paragraph 51. The applicant respectfully disagrees. Paragraph 51 of Brown reads, in pertinent part, as follows:

After the bounding DTD is completed, an XML bounding file can be written (for example, by a product development team) under the restrictions of the bounding DTD. A processing component (which, in preferred embodiments, operates within an editing component) then reads in this XML bounding file and, based on its specified restrictions, provides a set of editing actions that limit the user in what he can see and do with an XML element. This XML bounding file can then be changed, if desired, so that the processing component acts differently, depending on the revised restrictions in the XML bounding file. Thus, editing operations available in the same processing component (and on the same input XML file for which editing is to be bounded) can be tailored to different users with diverse needs or different tasks as long as there is a different implementation of the XML bounding file.

The cited paragraph discloses that the bounding file can be read to limit editing actions of the user. Additionally, the cited paragraph discloses that the editing restrictions can be tailored to different users. However, the cited paragraph fails to disclose or suggest "identifying a structural aspect of the structured electronic document that fails to conform to rules" as recited by the claim. The techniques Brown teaches operate to prevent any nonconforming aspects from being created in the electronic document. Consequently, in Brown the opportunity to identify such nonconforming aspects would not arise.

Furthermore, the cited paragraph does not disclose or suggest "suggesting one or more changes to a user which would correct the identified structural aspect." There is no mention in the cited paragraph that any suggestions are provided to a user.

The applicant respectfully submits that claim 1, as well as claims 2-12, which depend from claim 1, are in condition for allowance.

Claim 13 stands rejected over Brown. Claim 13 recites a computer-implemented method for validating and correcting a markup language document. A parent element of the markup language document is recursively validated. The validation includes validating attributes of the parent element, validating a content model of the parent element, and recursively validating one or more children of the parent element.

A structural aspect of the markup language document is identified as failing to conform to the rules associated with the markup language document. One or more changes are suggested to a user that would correct the identified structural aspect. User input is received selecting a

change of the suggested changes. The selected change is applied to the structured electronic document.

The examiner states that the validation of claim 13 is disclosed by paragraph 7 of Brown. The applicant respectfully disagrees. Paragraph 7 simply states that an XML parser can process an XML document type definition along with an XML document to validate the document as adhering to the data model defined by the document type definition. However, the cited paragraph does not disclose or suggest any validation process. Thus, the cited paragraph does not disclose or suggest a recursively validating an element of the markup language document.

Moreover, claim 13 recites identifying a structural aspect of the markup language document that fails to conform to rules associated with the markup language document. As discussed above, the cited sections of Brown do not disclose or suggest identifying a structural aspect as failing to conform to rules associated with the markup language document or suggesting changes to the user.

The applicant respectfully submits that claim 13, as well as claims 14-15, which depend from claim 13, are in condition for allowance.

Claim 16 stands rejected over Brown. Claim 16 recites a computer program product for correcting a structured electronic document that includes identifying a structural aspect of the structured electronic document that fails to conform to rules of a markup language. For the reasons set forth above with respect to claim 1, claim 16 as well as claims 17-27, which depend from claim 16, are in condition for allowance.

Claim 28 stands rejected over Brown. Claim 28 recites a computer program product for validating and correcting a markup language document. The computer program product includes instructions for recursively validating a parent element of the markup language document. For the reasons set forth above with respect to claim 13, claim 28 as well as claims 29-30, which depend from claim 28, are in condition for allowance.

The applicant respectfully requests that all pending claims be allowed.

By responding in the foregoing remarks only to particular positions taken by the examiner, the applicant does not acquiesce with other positions that have not been explicitly

Applicant : William Lin
Serial No. : 10/665,156
Filed : September 18, 2003
Page : 16 of 16

Attorney's Docket No.: 07844-610001 / P563

addressed. In addition, the applicant's arguments for the patentability of a claim should not be understood as implying that no other reasons for the patentability of that claim exist.

Please apply the charge for a one-month extension of time and any other charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: 31 July, 2006



Brian J. Gustafson
Reg. No. 52,978

Customer No.: 21876
Fish & Richardson P.C.
Telephone: (650) 839-5070
Facsimile: (650) 839-5071

50363140.doc



U.S. Serial No. 10/665,156
Reply to Action Mailed March 29, 2006
Attorney Docket No. 07844-610001
Annotated Sheet Showing Changes

110

<?xml version="1.0"?>
<!DOCTYPE greeting SYSTEM "hello.dtd">
<greeting>HELLO, world!</greeting>

FIG._1
Prior Art

210

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE greeting [
 <!ELEMENT greeting (#PCDATA)>
]>
<greeting>Hello, world!</greeting>

FIG._2
Prior Art

[Welcome to XMLSPY](#)
2004 Enterprise Edition

[User Manual](#)

[Introduction](#)

[Altova's XML Products](#)

[About XML](#)

[What is XML?](#)

[Why XML?](#)

[Adjunct Standards](#)

[User interface](#)

[Schema Editing](#)

[XML Editing](#)

[Authentic View](#)

[Project Management](#)

[Tutorial](#)

[Overview](#)

[Creating a schema from scratch](#)

[Creating a new Schema file](#)

[Adding elements to a schema](#)

[Adding elements with drag and drop](#)

[Configuring the Schema/WSDL view](#)

[Completing the basic schema](#)

[Making schema components reusable](#)

[Globals, extending simple and complex types](#)

[References, attributes and enumerations](#)

[Navigation shortcuts in schema documents](#)

[Generating Schema documentation](#)

[Creating an XML document](#)

[Creating and completing a new XML file](#)

[Editing in Text- and Grid view](#)

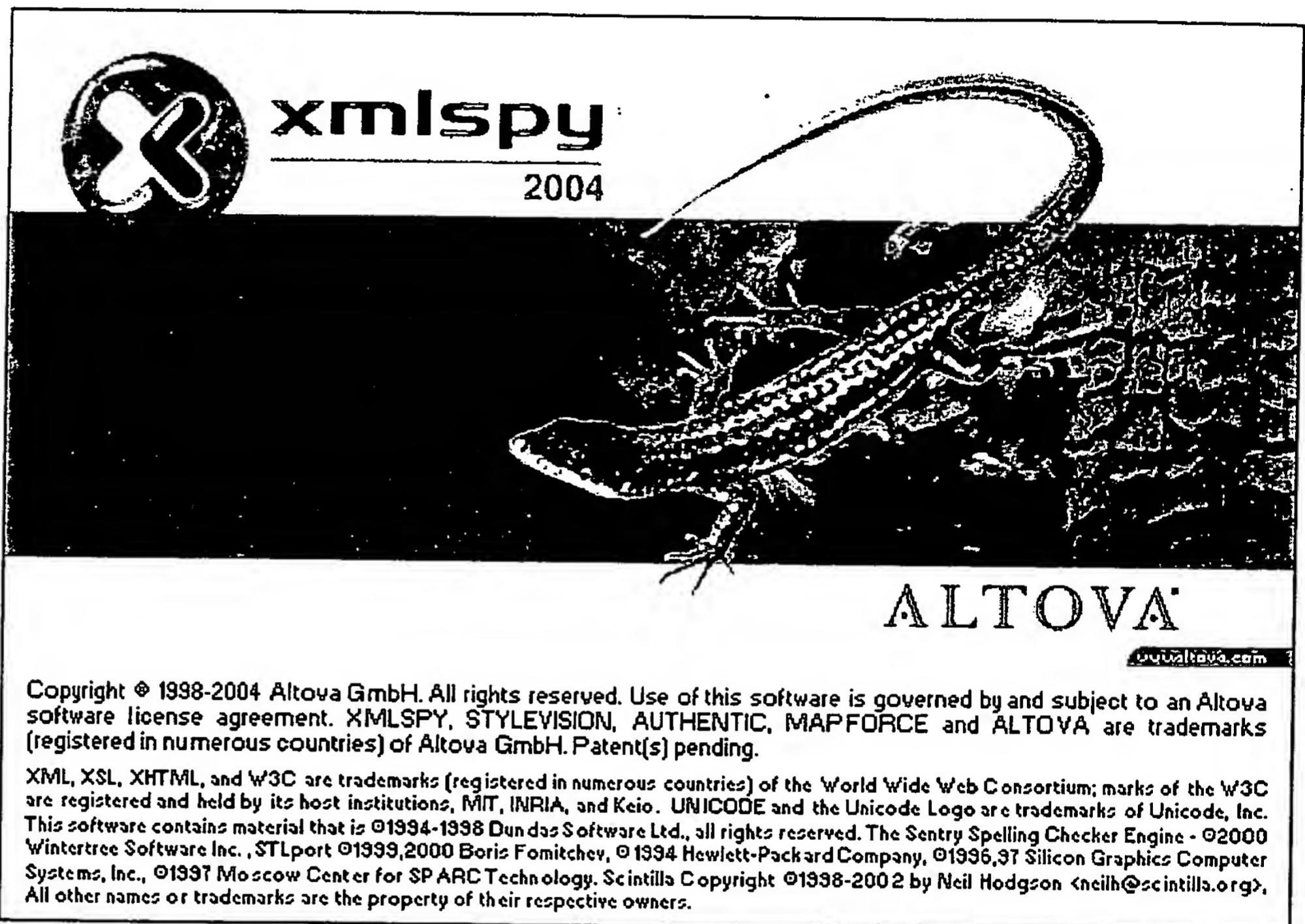
ALTOVA *markup your mind!*

www.altova.com

[Top](#) [Next](#)

Welcome to XMLSPY 2004 Enterprise Edition

xmlspy® 2004 is the industry standard XML Development Environment for designing, editing and debugging enterprise-class applications involving XML, XML Schema, XSL, XSLT, SOAP, WSDL and Web service technologies. It is the ultimate productivity enhancer for J2EE, .NET and database developers.



[Top](#) [Next](#)

© 2004 Altova

ALTOVA *markup your mind!*

www.altova.com

[Previous](#) [Top](#) [Next](#)

XML Editing

An XML document—and there are several kinds, including XML Schema, XSLT, XSL-FO—can be created and edited in Text View and Grid View. The screenshot below shows an XML document in Grid View with the Browser View of the XML below it.

The screenshot displays two windows side-by-side. The left window, titled 'CompanyLast.xml', is in 'Grid' view, showing an XML schema and data. The right window, titled 'XSL Output (F1)', is in 'Text' view, showing the generated XML output.

Grid View (CompanyLast.xml):

```

<!-- C. edited with XML Spy v4.0.1 U (http://www.xmlspy.com) by
Alexander Pilz (private) -->
<?xml version="1.0" encoding="UTF-8"?>
<Company>
  <xsi:schemaLocation> http://my-company.com/namespace AddressLast.xsd
  <xsi:type>US-Address</xsi:type>
  <Person>
    <Manager>1</Manager>
    <Degree>false</Degree>
    <Programmer>true</Programmer>
  </Person>
  <Person>
    <Manager>2</Manager>
    <Degree>true</Degree>
    <Programmer>false</Programmer>
  </Person>
</Company>
  
```

Text View (XSL Output (F1)):

```

<Your Company>
  <Name>US dependency</Name>
  <Street>Noble Ave.</Street>
  <City>Dallas</City>
  <State>Texas</State>
  <Zip>04812</Zip>
  <Person>
    <First>Alfred</First>
    <Last>Aldrich</Last>
    <Ext>33</Ext>
    <E-Mail>Aldrich@work</E-Mail>
    <Manager>false</Manager>
  </Person>
</Your Company>
  
```

Editing the document

In Grid View, you edit a document by (i) using the entry helpers to insert elements, attributes, etc (double-click an item in an entry helper to insert it), and (ii) then entering values for these nodes. In Text View, you can type in the name of elements, attributes, etc, and then enter values for these elements. You can use the entry helpers to insert elements and attributes (double-click an item in an entry helper to insert it). Text View also offers auto-complete features, such as completing the names of elements and attributes, adding closing tags, child nodes, and so on. You can set options for these features in the **Tools | Options** dialog.

Transforming the document

An XML document can be viewed as an HTML document in the Browser View if it is transformed to HTML. In order to transform an XML document to HTML you need an XSLT stylesheet. The XML document must therefore contain a reference to the XSLT stylesheet or an XSLT stylesheet must be assigned to that XML project. The transformation takes place implicitly when you click the Browser tab, and the HTML output of the transformation is displayed in the Browser View. If you wish to create an HTML output file, you must explicitly transform the XML document.

[Previous](#) [Top](#) [Next](#)

© 2004 [Altova](#)



www.altova.com

[Previous](#) [Top](#) [Next](#)

Validate XML

When you validate an XML document, XMLSPY 2004 first tries to locate any reference to a supported schema dialect within the document. It then loads the corresponding schema or DTD into memory, and uses its definitions to validate your XML instance document.

Once the Schema or DTD has been loaded into memory, XMLSPY 2004 also provides you with intelligent editing functions in the [Info Window](#) and [Entry-Helpers](#).

As long as the XML instance document is open, XMLSPY 2004 will also keep the Schema or DTD in memory (see the [Flush Memory Cache](#) command on the [DTD/Schema](#) menu).

[Previous](#) [Top](#) [Next](#)

© 2004 Altova

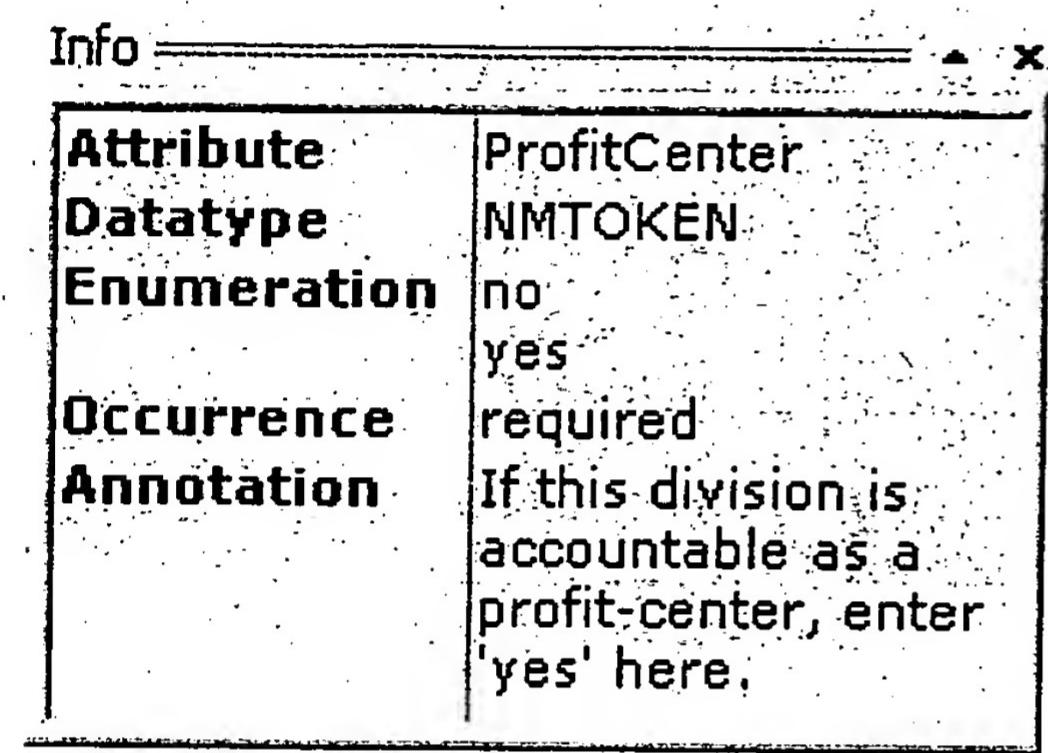
ALTOVA *markup your mind!*

www.altova.com

[Previous](#) [Top](#) [Next](#)

Info Window

In addition to the [entry helper](#) windows, XMLSPY 2004 provides a handy information window that shows the detailed information about the attribute or element that you are working on.



This information is available in all three editing views and can be a tremendous help in conjunction with the xsd:annotation feature of the new XML Schema draft, which allows a schema author to include comments or documentation on the use of each individual element or attribute into the schema itself.

[Previous](#) [Top](#) [Next](#)

© 2004 Altova



www.altova.com

[Previous](#) [Top](#) [Next](#)

Entry Helper Windows

XMLSPY 2004 helps you create valid XML documents by providing three palette-like windows that we call "entry helpers".

When you are working on an XML document based on a DTD or Schema, the built-in Intelligent Editing module constantly displays information on elements, attributes, and entities that can be inserted at the current cursor position, based on the information gathered from the Schema or DTD content model.

The entry helper windows have a XMLSPY 2004 prefix in Visual Studio .NET.

[Previous](#) [Top](#) [Next](#)

© 2004 Altova

[Previous](#) [Top](#) [Next](#)

Validate

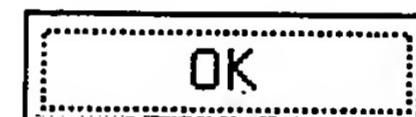
This command validates an XML document against the rules set forth in its Schema or Document Type Definition (DTD) or it can validate any XML Schema or DTD against the rules set forth in the corresponding specification.

XMLSPY 2004 uses its built-in incremental validating parser that supports all major Schema dialects, such as DTD, DCD, XML-Data, and the new W3C XML Schema Definition Language (XSD).

If the validation succeeds, a brief message is displayed at the bottom of the main window:



This file is valid.



If an error is encountered during the validation, the source of the problem is highlighted and a corresponding error message is shown:



This file is not valid:

Mandatory element 'content' expected in place of 'table'



The Validate command also automatically includes a well-formedness check, so there is no need to first use the Check well-formedness command manually before validating a file.

Please note that you can use this command to validate XML instance documents and for validating XML schemas or DTDs.

This command normally operates on the active main window

[Previous](#) [Top](#) [Next](#)

© 2004 Altova



www.altova.com

Altova to Showcase Newly Released XMLSPY 2004 and MAPFORCE 2004 at OracleWorld

ALSO AT ORACLEWORLD; ALTOVA PRESIDENT AND CEO, ALEXANDER FALK TO DELIVER TECHNICAL PRESENTATION ON ARCHITECTING XML SCHEMAS FOR ORACLE XML DATABASES

BEVERLY, Mass., September 5, 2003 – Altova Inc. (www.altova.com) producer of XMLSPY, the leading XML development environment with over one million registered users worldwide, today announced that at next week's OracleWorld Conference and Expo it will be conducting the first public demonstrations of the company's recently announced 2004 line of XML development tools including XMLSPY 2004 and the company's newest product MAPFORCE 2004.

In addition to showcasing its newest product line, Altova's President and CEO, Alexander Falk, will deliver a technical presentation at OracleWorld, entitled "Architecting XML Schemas for Oracle XML Databases." Mr. Falk's one hour presentation will be held in room 130 of the Moscone Convention Center on Wednesday, September 10 at 8:30 a.m.

Altova engineers will be available for product demonstrations during OracleWorld at Booth 2018. The OracleWorld Conference and Expo is taking place next week, September 8-11 at the Moscone Convention Center in San Francisco.

Altova's newest product offering, MAPFORCE 2004, is a powerful, visual data integration tool, which auto-generates custom data mapping code in multiple output languages such as XSLT and Java, to enable programmatic XML-to-XML or database-to-XML data transformations. Altova's new MAPFORCE 2004 provides a powerful, 2-step XML-based approach to enterprise data integration. Using MAPFORCE 2004, data architects can programmatically convert data into XML from any database, simply by drawing visual mappings from relational databases to any data model expressed in XML schema. MAPFORCE 2004 will then auto-generate the software program code required to programmatically marshal data from the source database to the target XML schema. Next, data can easily be transformed from one XML format to another, by visually drawing mappings between different XML schema data models. The MAPFORCE 2004 approach to enterprise data integration ensures compatibility and interoperability across different platforms, servers, programming languages and database environments.

Altova has added numerous new features to its award-winning product family of XML development tools and content authoring tools. The key new features of XMLSPY 2004 include:

- Microsoft Visual Studio® .NET® Integration – An added convenience for Microsoft Developers, enabling them to use the XMLSPY directly within their preferred editing environment.
- XML Differencing – Visually compare files and directories in a highly configurable, XML-aware manner, which takes into account such variables as namespaces.

- XPath 2.0 Analyzer – Experiment with a beta implementation of the W3C's XPath 2.0 working draft of May, 2003.
- Completely re-implemented Database Utilities – Native support for Oracle databases, auto-generation of Microsoft SQL Server and Oracle 9i XML Schema extensions, generation of XML Schemas according to SQL/XML standard.

Pricing and Availability

XMLSPY 2004 is available immediately in three distinct editions: Enterprise Edition, Professional Edition, and Home Edition, available for purchase from the Altova Online Shop at <http://www.altova.com/order>, with prices for a single-user license starting at \$990, \$399, and \$49, respectively. A free 30-day trial download is available from <http://www.altova.com/download>.

MAPFORCE 2004 is available for a free 30-day trial download from <http://www.altova.com/download>, or purchase for \$499 for a single user license from the Altova Online Shop.

Special Introductory Offers

Altova is making available several introductory offers, beginning on September 2, 2003 and running until November 30, 2003.

- Every customer who purchases a new license of XMLSPY 2004 Enterprise Edition or upgrades from a previous product to XMLSPY 2004 Enterprise Edition receives MAPFORCE 2004 for free.
- Any Altova customer who upgrades their existing Altova software products to the corresponding new 2004 products will receive a special introductory price of up to 25% off the regular upgrade price, please see the Altova Online Shop for a complete pricelist and additional details.
- Existing XMLSPY 5 Professional Edition customers may upgrade to XMLSPY 2004 Enterprise Edition, at a special introductory upgrade price of \$499!

Altova's XMLSPY 2004 Road Show - Coming to a City Near You!

Don't miss the first-ever special technical presentation and informational session about Altova's new 2004 product line of XML development and XML content authoring tools, featuring "The XMLSPY Handbook" author, Larry Kim. All participants will receive a free Altova T-shirt. The session will cover topics including: What's new in XMLSPY 2004, data integration with MAPFORCE 2004, building XML content editing applications with AUTHENTIC 2004, and a question & answer session.

When & Where:

- * San Francisco: Monday, September 22, from 3:30-5 pm
- * Washington, DC: Tuesday, September 30, from 3:30-5 pm
- * Boston: Thursday, October 2, from 3:30-5 pm

For complete locations and directions and to register for free, please visit: <http://www.altova.com/register>.

About Altova

Altova is a leading provider of XML software tools, components, and professional service solutions, including XMLSPY 2004, the industry standard XML Development Environment. Altova, with offices in Beverly, Mass. and Vienna, Austria, was founded in 1992 and has been actively involved in the XML market from the early conception of Extensible Markup Language. Altova's 2004 product line of XML development tools and components is the leading choice of Fortune 500 and Global 1000 companies. Altova's XMLSPY is the world's best-selling XML tool and has won the leading industry awards including PC Magazine's Editor's Choice Award and Best Product of 2002. Altova is a technology partner with the world's leading software companies including Microsoft, Oracle, BEA, Borland, IONA, and Software AG. Altova is a member of the W3C and WS-I. Visit Altova on the Web at <http://www.altova.com>.

###

Altova, XMLSPY, STYLEVISION, AUTHENTIC and AXAD are trademarks of Altova GmbH registered in numerous countries. The names of and reference to companies and products mentioned herein may be the trademarks of their respective owners.

For more information:

Paul Roberts
Public Relations Manager, Altova Inc.
978-816-1567
paul.roberts@altova.com

For more information, sales only:

sales@altova.com
USA: (978) 816-1600
EU: +43 (1) 545 5155 - 0

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.